

REMARKS

Claims 100-117 have been amended. Claims 1-24, 51-73, 100-117, 136 and 138 are pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Allowable Subject Matter:

Previously, in the Final Office Action, the Examiner stated that claims 8-11, 57-60 and 106-109 were objected, but otherwise allowable if rewritten in independent form. In the present Office Action, the Examiner has failed to provide a prior art rejection for any of these claims. As such, Applicants' presume that claims 8-11, 57-60 and 106-109 are objected to for being dependent upon a rejected based claim, but would otherwise allowable if rewritten in independent form, as previously noted by the Examiner.

Section 101 Rejection:

The Examiner rejected claims 100-117 under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Applicants respectfully traverse this rejection. However, in order to expedite prosecution, claims 100-117 have been amended. Removal of the § 101 rejection is respectfully requested.

Section 112, First Paragraph, Rejection:

The Examiner rejected claims 1-24, 51-73 and 100-117 under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement. Applicants respectfully traverse this rejection for at least the reasons presented below.

Specifically, the Examiner contends that claims 1-24, 51-73 and 100-117 “[fail] to comply with the written description requirement” and asserts, “[Applicants’] specification fails to define ‘appears.’” (Office Action, dated October 19, 2007, p. 3).

However, there is no requirement that applicants' specifically or explicitly define every word or phrase used in the claims. Instead, quite to the contrary, "the words of the claim must be given their plain meaning *unless* applicant has provided a clear definition in the specification" (emphasis added, M.P.E.P. 2111.01) *In re Zletz*, 893 F.2d 319, 321, 13 USPQ2s 1320, 1322 (Fed. Cir. 1989). It is thus clearly improper to reject claims 1-24, 51-73 and 100-117 merely because they include a term not specifically or explicitly defined in Applicants' specification.

Furthermore, as shown below, Applicants' specification provides proper support for all of Applicants' claims, including limitations using the term, "appears". As a non-limiting example, Applicants' specification describes an embodiment in which a bridge service may be looked up and instantiated by a client "just as any other service in the distributed computing environment" and that the bridge service may interact with, and execute, other services, such as an enterprise service. The specification further describes that, to the client, "the bridge service **appears** to be the only service" and that "the enterprise service is hidden to the client" (p. 148). Thus, support for the term "appears" is found within Applicants' specification. Applicants' specification also describes embodiments in which various entities, including devices and services (among others), "may be bridged into the distributed computing environment using proxy services" (p. 149). The plain meaning of the terminology used in Applicants' specification and claims is completely clear to any one of ordinary skill in the art.

Section 112, Second Paragraph, Rejection:

The Examiner rejected claims 1-24, 51-73 and 100-117 under 35 U.S.C. § 112, second paragraph, as being indefinite. Applicants respectfully traverse this rejection for at least the reasons presented below. Specifically, the Examiner argues that these claims "recite the limitations, 'appears'" (Office Action, p. 3). However, mere use of a particular word or phrase does not imply that a claim is indefinite. Additionally, the Examiner has not explained how or why use of the word, 'appears' makes claims 1-24, 51-73 and 100-117 indefinite.

Applicants further note that the “test definiteness under 35 U.S.C. 112, second paragraph, is whether ‘those skilled in the art would understand what is claimed when the claim is read in light of the specification’” and that “Breadth of a claim is not to be equated with indefiniteness.” M.P.E.P. 2173.02, 2173.04; *In re Miller*, 441 F.2d 689, 169 USPQ 597 (CCPA 1971).

As noted in § 2173.05(e) of the M.P.E.P., “[i]f the scope of a claim would be reasonable ascertainable by those skilled in the art, then the claim is not indefinite.” Applicants submit that one skilled in the art would be able to ascertain the scope of Applicants’ claims. Applicants’ claims recite that a proxy service provides an interface to, and “appears as” a second entity to a first entity. One skilled in the art would clearly understand the scope of Applicants’ claim.

Double Patenting Rejection:

The Examiner rejected claims 1-24, 51-73, 100-117, 136 and 138 under the judiciary created doctrine of obviousness-type double patenting as being unpatentable over claims 1-45 of U.S. Pat. No. 6,868,447. Applicants respectfully traverse this rejection for at least the reasons below.

As argued in Applicants’ recent Appeal Brief, **the Examiner has failed to state a *prima facie* obviousness-type double patenting rejection.** According to MPEP 804.II.B.1, to state a proper obviousness-type double patenting rejection, the Examiner should list the differences between each rejected claim and the claims of the other patent/application, and for each difference the Examiner should give the reasons why a person of ordinary skill in the art would conclude that the invention defined in the claim at issue would have been an obvious variation of the invention defined in a claim of the other patent/application. Simply noting a few similarities between the claims does not satisfy the Examiner’s burden to state valid reasons (supported by evidence of record) why a person of ordinary skill in the art would conclude that the invention defined in the

claim at issue would have been an obvious variation of the invention defined in a claim of the other patent/application. Nor has the Examiner specifically addressed **each difference** of the claim of the present application compared to the claim of the other application.

Instead, the Examiner merely states, “[a]lthough the conflicting claims are not identical, they are not patentably distinct from each other because the patent teaches the limitations as disclosed such that the interpretation of a first entity accessing a second entity through messages in a data representation language is equivalent to a first client sending a first message in to a first service and the first service generating a set of results in response to the first message, wherein the set of results are expressed in a data representation language and using a space, advertisement, XML, and URI.” The Examiner has not addressed every difference between the claims. The Examiner clearly has not met the requirements (as stated in MPEP 804.II.B.1) to establish a *prima facie* obviousness-type double patenting rejection.

The Examiner does note that the claims of U.S. Patent No. 6,868,447 do not include the limitations in regard to bridging and proxy service as recited in the claims of the present application. The Examiner contends that these limitations were well known and would be obvious and refers to the Tuatini (U.S. Patent Application Publication No. 2002/0032783) for support. Applicants traverse the Examiner’s assertion that these limitations were well known in the context of Applicants’ claimed invention. Additionally, the Examiner’s reliance on Tuatini for the double patenting rejection (and the § 102 and § 103 rejections discussion below) is misplaced because i) Tuatini fails to teach or suggest the particular limitations of Applicants’ claims (as discussed below regarding individual claim rejections), and ii) Tuatini is prior art (as discussed below). As to the several other references cited by the Examiner, these references may indicate that bridging and proxy services were well known *in other contexts*, but fail to show that bridging and proxy services were well known in the context of the particular limitations of Applicants’ claimed invention. Also, the cited references do not teach or suggest bridging and proxy services that function as recited in Applicants’ claims

The Examiner has incorrectly stated that Applicants' previous argument was bridging and proxy service is *not well known in the art* (See, Advisory Action). The Examiner has misstated Applicants' argument. Applicants are arguing that a bridging and proxy service was not well known *in the context of* Applicants' *claimed invention*. Bridging and proxy services may have been well known in other contexts, but Applicants assert that bridging and proxy services were not well known that function as recited in Applicants' claims and in the context of Applicants' claimed invention which recites a *specific combination* of features. Nor has the Examiner stated a reason to include such bridging and/or proxy services in the context of Applicants' claimed invention.

For at least the reasons above, Applicants respectfully request removal of the double patenting rejection.

Section 102(e) Rejections:

The Examiner rejected claim 138 under 35 U.S.C. § 102(e) as being anticipated by Tuatini (U.S. Patent Application Publication No. 2002/0032783), claims 1, 51, 100, 136 and 138 under 35 U.S.C. § 102(e) as being anticipated by Zintel (U.S. Patent 7,130,895), and claims 1, 51, 100, 136 and 138 under 35 U.S.C. § 102(e) as being anticipated by Bowman-Amuah (U.S. Patent 6,842,906). Applicants respectfully traverse these rejections for at least the following reasons.

Regarding the rejection of claim 138 as anticipated by Tuatini, the rejection is improper because (among other reasons), Tuatini is not prior art. The Examiner has the burden of proof to produce the factual basis for the rejection. *In re Warner*, 154 USPQ 173, 177 (C.C.P.A. 1967), *cert. denied*, 389 U.S. 1057 (1968). The Tuatini application was filed on January 2, 2001, after Applicants' filing date of October 19, 2000. Tuatini does claim the benefit of provisional applications filed December 30, 1999. However, the December 30, 1999 filing date can only be used as Tuatini's 35 U.S.C. § 102(e) prior art date for the subject matter that is common to both the Tuatini

patent and one of the provisional applications. From even a cursory review it is clear that Tuatini's published application differs greatly from its provisional applications. The Examiner has not shown that all of the subject matter from Tuatini's published application relied upon in the rejection is found in one of Tuatini's provisional applications.

Moreover, the Tuatini publication is not entitled to the December 30, 1999 date as a section 102(e) prior art date unless at least one claim of the Tuatini publication is supported (under 35 U.S.C. § 112) in the provisional application. Under 35 U.S.C. 119(e)(1), a published patent application is not entitled to its provisional application's filing date as a prior art date unless at least one claim of the published application is supported (per 35 U.S.C. § 112) in the provisional application. The rejection is improper unless the Examiner can show that Tuatini's published application has the necessary claim support in one of the provisional applications to be entitled to the provisional application's filing date as its § 102(e) prior art date. *See also* M.P.E.P. § 2136.03(IV). The Examiner has not made such a showing.

In response to Applicants' previous request that the Examiner particularly point out those portions of Tuatini's provisional application relied upon by the Examiner, as required by 37 CFR 1.104(c)(2), the Examiner refers to pages 4, 16, 78, 112, 236, 324, and 428 of Tuatini's provisional application 60/173,712. **However, none of these pages support the subject matter relied upon by the Examiner in the rejection of claim 138.** Three of these pages (78, 112, 136) are portions of a general, high level, overview and do not support, or have any relevance to, the portions of Tuatini's publication that the Examiner is relying upon in his rejection. For example, page 78 is merely a title page that mentions "GEPS E-Business Architecture Workout" and "Java Coding Standards". Page 112 is an illustration depicting various categories, such as auditing, security, event logging, etc. illustrated as puzzle pieces with an "application component" puzzle piece in the center. Page 236 lists various developmental steps, such as "define and develop partitioning, replication and referral policies", "develop load, maintenance and synchronization services", "develop framework LDAP query components", etc. Nothing

in these three pages supports the specific subject matter of Tuatini relied upon by the Examiner in his rejections. Page 4 illustrates the logical connects between various components that collaborate with Tuatini's ControlServlet component. Page 16 describes various Java classes, such as the Application, BaseGenericServlet and BaseHttpServlet classes. Page 428 provides a discussion of Tuatini's messaging component and is the only page referred to by the Examiner that teaches something relied upon by the Examiner, namely Tuatini's messaging component. The pages referred to by the Examiner, whether considered individually or as a group, do not support hardly any of the subject matter of Tuatini relied upon by the Examiner in his rejection.

Page 324 of Tuatini's provisional application 60/173,712 illustrates a flowchart. However, the details of the flowchart are completely illegible. Thus, page 324 (as it is illegible) does support any of the subject matter relied upon by the Examiner in his rejection. In response to Applicants' earlier argument that page 324 is illegible, the Examiner additionally cites pages 305 – 323 of Tuatini's provisional application. These pages first describe pseudo-code of various class definitions, such as orderquery and order, in Tuatini's system as well as various aspects of Tuatini's schema compiler, such as the serialize and deserialize functions, the overall processing of the parser component, and the code generator component. The pages also describe the processing of a service request that is passed and may return serialized messages. **However, as with the other pages of the 60/173,712 provisional application cited by the Examiner, none of these pages support the subject matter relied upon by the Examiner in the rejection of claim 138.**

The Board of Patent Appeals and Interferences recently discussed this very issue in the Decision on Appeal, U.S. Patent Application No. 10/054,809, August 21, 2007, p. 8, lines 1-7 and 11-20. The board stated that the Examiner's "rejection should show, to establish a *prima facie* case for unpatentability, where support resides in the earlier provisional application for each instance of specific subject matter relies upon in the published application, including an explanation why the provisionals would still be recognized by the artisan as providing support if not 'word for word' the same as the later

text or drawings" (emphasis added). The board further pointed out that "Mere reference to the text or drawings ... is not sufficient." *See*, Decision on Appeal, U.S. Patent Application No. 10/054,809, August 21, 2007, p. 8, lines 1-7 and 11-20).

This is exactly the situation in the instant case, namely that the Examiner has failed to provide a *prima facie* rejection because the Examiner has not shown that Tuatini's provisional application fully supports every instance of the subject matter of Weisman's later utility application relied on by the Examiner.

Moreover, the pages of the Tuatini provisional application cited by the Examiner do not support any of the claims of Tuatini's published application. For instance, the pages cited by the Examiner do not support claim 1 of the Tuatini application per the requirements of 35 U.S.C. § 112, first paragraph. The rejection is improper unless the Examiner can show that Tuatini's published application has the necessary claim support in the provisional application to be entitled to the provisional application's filing date as its § 102(e) prior art date. *See also* M.P.E.P. § 2136.03(IV). The Examiner has not met his burden of proof to show that Tuatini qualifies as prior art for this reason as well.

In further regard to claim 138, regardless of whether or not Tuatini qualifies as prior art, Tuatini does not disclose the proxy service providing to the first entity an interface to a second entity in the second computing environment, wherein providing an interface comprises sending to the first entity a schema defining one or more messages in the data representation language for accessing the second entity. The Examiner cites paragraphs [0166-0168] where Tuatini describes the use of an LDAP directory service. However, Tuatini does not teach sending a client application (which the Examiner equates to a first entity) a schema defining one or more messages in the data representation language for accessing the second entity. Instead, Tuatini describes how a LDAP directory may include a schema defining object classes of information that can be stored in the directory entries. Tuatini does not mention anything about a schema defining messages in a data representation language for accessing the LDAP directory

service, as would be required for the Examiner’s line of reasoning to be correct. Tuatini also fails to mention sending such a schema to the client application. Furthermore, Tuatini teaches how a client accesses a LDAP directory by instantiating a directory manager object and uses methods of the directory manager object to retrieve other objects (both directory entry objects and adapter objects) for accessing particular directory entries. (Tuatini, paragraph [0167]). **Thus, Tuatini’s system provides access objects for the entries of a LDAP directory service rather than sending a schema defining messages in a data representation language for accessing the directory service.**

The Examiner also cites paragraphs [0122-0132] where Tuatini describes the use of a XML document type definition (DTD) to specify message parameters used to request service functions. However, the cited passage does not mention a proxy service sending the XML DTD to a client component, which would be required if the Examiner’s interpretation of Tuatini were correct. Instead, Tuatini describes that the XML DTD may be a part of a group of information for each shared service providing functionality to clients and that the information is “made available to others” (Tuatini, paragraph [0125]). The mere statement that an XML DTD is *made available* to others does not disclose the specific limitation of *a proxy service sending* a schema to a first entity, as recited in Applicants’ claim 138. There are, in fact, many ways in which information may be “made available” to entities in a distributed computing environment, as is well known in the art. For example, Tuatini states that the XML DTDs may be stored separately from the access interface information (Tuatini, paragraph [0128]) and that Tuatini’s messaging component may *retrieve* the XML DTD to verify that a message is properly formatted, thus implying that in Tuatini’s system the XML DTDs are made available by storing them in a shared location.

The Examiner has also incorrectly characterized (see, Advisory Action) Applicants’ previous argument by stating, “it seems that Applicant is implying that the cited references do not disclose ‘proxy client’ / ‘accessing of a Jini Service proxy from clients’, which the claims are not limited to.” As stated above (and previously) Applicants argue, in part, that Tuatini does not disclose a proxy service providing to the

first entity an interface to a second entity ..., wherein providing an interface comprises sending to the first entity a schema defining one or more messages in the data representation language for accessing the second entity. The Examiner cites Tuatini's teachings regarding the use of a XML DTD. However, as stated quite clearly above, Tuatini does not disclose a proxy service sending the XML DTD, which the Examiner equates to the schema of Applicants' claim, to Tuatini's client component, which the Examiner equates to the first entity of Applicants' claim. Instead, as noted above, Tuatini merely teaches that the XML DTD (schema) is *made available* and that Tuatini's messaging component may retrieve the XML DTD (schema). Merely making a XML DTD available such that it may be retrieved does not disclose the proxy service sending to the first entity a schema. Thus, Tuatini clearly fails to disclose the proxy service providing to the first entity an interface to a second entity in the second computing environment, wherein said providing an interface comprises sending to the first entity a schema defining one or more messages in the data representation language for accessing the second entity.

As anticipation under 35 U.S.C. § 102 requires that the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim (*Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984)), Tuatini clearly does not anticipate Applicants' claim 138. Thus, the rejection of claim 138 is not supported by the prior art and removal thereof is respectfully requested.

As noted above, the Examiner rejected claims 1, 51, 100, 136 and 138 under 35 U.S.C. § 102(e) as being anticipated by Zintel (U.S. Patent 7,130,895). Applicants respectfully traverse this rejection for at least the reasons below.

Regarding claim 1, contrary to the Examiner's assertion, **Zintel fails to disclose a first entity in the first computer environment accessing a proxy service through messages in a data representation language.** The Examiner cites col. 6 of Zintel,

without citing or referring to any particular part of Zintel’s system. Zintel teaches a device control model that provides an integrated set of processes (e.g., addressing, naming, discovery and description processes) that enables self-setup by devices to interoperate with other devices on a network (Abstract). Zintel’s system relies heavily on Universal Plug and Play (UPnP) and col. 6, cited by the Examiner, is part of a description of UPnP. However, the cited passage does not mention anything about an entity accessing a proxy service through messages in a data representation language.

In fact, Zintel fails to teach anything about accessing a proxy service through messages in a data representation language. In contrast, Zintel teaches the use of proxy object that are accessed via a particular and specific set of API calls that Zintel describes in some detail. For instance, Zintel describes a rehydrator 410 that “operates as a universal adaptor to provide a programmatic interface to any service-specific protocol of a remote computing device” and that rehydrator 410 “exposes a suitable API to applications” (Zintel, col. 22, lines 16-21 and 36-56).

Rather than being accessed through messages in a data representation language, Zintel’s rehydrator is instantiated, invoked and controlled via programmatic API functions. For instance, Zintel teaches that in a preferred embodiment rehydrator 410 is “an internal Microsoft Windows component that routes service control requests from the UPnP API to devices” and that “the Rehydrator performs a mapping between API calls and network protocols” (Zintel, col. 23, lines 24-41). Zintel also describes a specific set of API methods used to interact with rehydrator 410 (e.g., RehydratorCreateServiceObject, RehydratorQueryStateVariable, RehydratorGetServiceProperty, RehydratorInvokeServiceAction, etc., col. 23-26).

Thus, not only does Zintel fail to disclose anything about an entity accessing a proxy service through messages in a data representation language, Zintel teaches a completely different method for accessing a proxy service that does *not* involve messages in a data representation language.

In further regards to claim 1, Zintel also fails to disclose **the proxy service providing to the first entity an interface to a second entity in the second computing environment, wherein the proxy service appears to the first entity as the second entity.** The Examiner cites col. 8 of Zintel. However, the cited passage does not describe a proxy service that appears to be a first device to a second device, as recited in Applicants' claim. Instead, the cited passage is part of a larger "terminology" section and discusses various types of modules and devices in Zintel system. The Examiner is apparently relying on Zintel's teachings regarding bridging, bridged devices, and bridging services. However, Zintel's bridging does not include, nor does Zintel disclose, anything about a proxy service appearing to a first entity as a second entity.

To the contrary, Zintel teaches a system in which a bridge or proxy service does *not* appear to a first entity as a second entity. Zintel's proxies and bridges, such as rehydrator 410, map API functions to network protocols. In fact, Zintel specifically and expressly teaches that "the bridge itself *exposes* a collection of UPnP Controlled Devices to User Control Points" and that "Bridged devices are UPnP compliant and are *exposed* to other UPnP devices through a UPnP Bridge" (emphasis added, col. 8, lines 14-25 and lines 30-38). Clearly a bridge that exposes the controlled device cannot be said to appear to a first entity as the second entity, as recited in Applicants' claim.

Furthermore, Zintel teaches that when using a proxy or bridge, such as rehydrator 410, applications pass information regarding a desired service object to be accessed via rehydrator 410. for instance, Zintel teaches that applications can query service properties invoking a GetProperty method that "makes a call to the RehydratorQueryStateVariable() function" (col. 24, lines 34-49). Zintel describes how the parameters of this function "supply the service instance specific information," the "Service Type Identifier," and "the name of the variable that is being queried" that the rehydrator will validate "against its internal list of state variables exported by the service" (col. 24, lines 49-60). Thus, Zintel expressly teaches invoking a particular and specific function of bridge's exposed API (e.g., RehydratorQueryStateVariable) passing information about a particular bridged or

proxied service. Zintel's bridges, such as rehydrator 410, do not appear to the first entity as the second entity, contrary to the Examiner's contention.

Applicants remind the Examiner that anticipation requires the presence in a single prior art reference disclosure of each and every limitation of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The **identical invention must** be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As discussed above, Zintel fails to disclose **a first entity in the first computer environment accessing a proxy service through messages in a data representation language and wherein the proxy service appears to the first entity as the second entity**. Therefore, Zintel cannot be said to anticipate claim 1.

Thus, for at least the reasons presented above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 51 and 100.

Regarding claim 136, contrary to the Examiner contention, **Zintel fails to disclose a first entity in the first computing environment accessing a proxy service through messages in a data representation language**. The Examiner cites col. 6 of Zintel. However, as noted above regarding claim 1, Zintel fails to disclose accessing a proxy service through messages in a data representation language. Instead, Zintel teaches accessing a proxy or bridge service, such as rehydrator 410, via API functions, not through messages in a data representation language. Zintel teaches the use of proxy object that are accessed via a particular and specific set of API calls that Zintel describes in some detail. For instance, Zintel describes a rehydrator 410 that "operates as a universal adaptor to provide a programmatic interface to any service-specific protocol of a remote computing device" and that rehydrator 410 "exposes a suitable API to applications" (Zintel, col. 22, lines 16-21 and 36-56).

Rather than being accessed through messages in a data representation language, Zintel's rehydrator is instantiated, invoked and controlled via programmatic API functions. For instance, Zintel teaches that in a preferred embodiment rehydrator 410 is “an internal Microsoft Windows component that routes service control requests from the UPnP API to devices” and that “the Rehydrator performs a mapping between API calls and network protocols” (Zintel, col. 23, lines 24-41). Zintel also describes a specific APIs (e.g., RehydratorCreateServiceObject, RehydratorQueryStateVariable, RehydratorGetServiceProperty, RehydratorInvokeServiceAction) to interact with proxies and bridges.

In further regard to claim 136, contrary to the Examiner’s contention, **Zintel fails to disclose wherein the proxy service providing to the first entity an interface to a second entity in the second computing environment comprises providing an advertisement for the second entity, wherein the advertisement for the second entity includes access information for accessing the second entity in the second environment from the first environment.** The Examiner cites col. 9 of Zintel. The cited passage is part of a larger “terminology” section. Presumably the Examiner is referring to Zintel’s service “description document”. However, even a cursory reading of Zintel demonstrates that Zintel’s proxy service does not provide the description document to the proxy service’s client. In contrast, Zintel expressly teaches that the proxy service, such as the rehydrator, obtains the description document from the client.

For instance, Zintel teaches that the description document is passed as a parameter to RehydratorCreateServiceObject() API call (col. 23, line 58- col. 24, line 11). Zintel also states that a “controlled Device or Bridge must be able to describe *to a User Control Point* the protocols required to control its Services” using the Description Document and that once “the Description Document is uploaded *into the User Control Point* the Rehydrator 410 can extract the SCPD from it” (emphasis added, col. 22, lines 6-15). Thus, Zintel fails to disclose where the proxy service providing an interface to a second entity comprises providing an advertisement for the second entity, where the advertisement includes access information for accessing the second entity.

Zintel clearly fails to disclose all the limitations of, and thus cannot be said to anticipate, claim 136. The rejection of claim 136 is not supported by the cited art and removal thereof is respectfully requested.

Regarding claim 138, **Zintel fails to disclose a first entity in the first computing environment accessing a proxy service through messages in a data representation language.** As noted above regarding claim 1, Zintel fails to disclose accessing a proxy service through messages in a data representation language. Instead, Zintel teaches accessing a proxy or bridge service, such as rehydrator 410, via API functions, not through messages in a data representation language. Zintel teaches the use of proxy object that are accessed via a particular and specific set of API calls that Zintel describes in some detail. For instance, Zintel describes a rehydrator 410 that “operates as a universal adaptor to provide a programmatic interface to any service-specific protocol of a remote computing device” and that rehydrator 410 “exposes a suitable API to applications” (Zintel, col. 22, lines 16-21 and 36-56).

Rather than being accessed through messages in a data representation language, Zintel’s rehydrator is instantiated, invoked and controlled via programmatic API functions. For instance, Zintel teaches that in a preferred embodiment rehydrator 410 is “an internal Microsoft Windows component that routes service control requests from the UPnP API to devices” and that “the Rehydrator performs a mapping between API calls and network protocols” (Zintel, col. 23, lines 24-41). Zintel also describes a specific APIs (e.g., RehydratorCreateServiceObject, RehydratorQueryStateVariable, RehydratorGetServiceProperty, RehydratorInvokeServiceAction) to interact with proxies and bridges.

In further regard to claim 138, **Zintel also fails to disclose the proxy service providing to the first entity an interface to a second entity in the second computing environment, wherein said providing an interface comprises sending to the first entity a schema defining one or more messages in the data representation language**

for accessing the second entity. The Examiner cites col. 9 of Zintel. However, the cited passage makes no mention whatsoever about a proxy service sending to a first entity a schema defining one or more messages in a data representation language for accessing the second entity. Presumably, the Examiner is relying on Zintel's description document. However, as shown above regarding claim 136, Zintel's proxy services, such as the rehydrator, do not send the description document, or any other schema defining messages in a data representation language, to Zintel's User Control Point applications. Instead, Zintel teaches, as noted above regarding claim 136, that the user control point applications provide the service description document (DCPD), including the Service Control Protocol Definition (SCPD) to the proxy service, such as via the pDCPD parameter of a create service object API call, such as the RehydratorCreateServiceObject() API call.

Thus, Zintel's proxy service cannot be said to include or disclose the proxy server sending to the first entity a schema defining one or more messages in the data representation language for accessing the second entity, as recited in Applicants' claim.

For at least the reasons above, the rejection of claim 138 is not supported by the cited art and removal thereof is respectfully requested.

As noted above, the Examiner rejected claims 1, 51, 100, 136 and 138 under 35 U.S.C. § 102(e) as being anticipated by Bowman-Amuah (U.S. Patent 6,842,906). Applicants respectfully traverse this rejection for at least the reasons below.

Regarding claim 1, contrary to the Examiner assertion, **Bowman-Amuah fails to disclose a first entity in the first computing environment accessing the proxy service through messages in a data representation language.** Bowman-Amuah teaches a system that interfaces a client with a naming service allowing access to a plurality of different sets of services from a plurality of globally addressable interfaces. Bowman-Amuah teaches proxies are generated based on the received locations of global

addressable interfaces and that the proxies are maintained in an allocation queue and in a proxy pool. The Examiner cites col. 43, presumably relying on Bowman-Amuah's browser extension services and plug-ins. However, Bowman-Amuah's browser extension services and plug-ins do not involve, nor disclose, accessing a proxy service through messages in a data representation language.

Firstly, browser extensions and plug-ins are not proxy services and are not accessed through messages in a data representation language. A browser may communicate to remote servers, using HTTP, but is not accessed as a proxy service through messages in a data representation language. Furthermore, Bowman-Amuah's proxy services are not described as being accessible through messages in a data representation language.

Instead, Bowman-Amuah's proxies appear to be accessed via a set of API calls. For instance, Bowman-Amuah teaches that proxies are stored in various pools, such as a ProxyPool and an AllocationPool. Bowman-Amuah teaches a set of C++ classes and templates for creating, storing and accessing proxies. Specifically, Bowman-Amuah teaches that a proxy is based on a PooledProxy that "is the base class for the pooled proxy" that "acts as a wrapper for a Proxy" (col. 232, lines 1-53). Bowman-Amuah further teaches that "Clients who wish to use a pooled proxy will create a handle as a wrapper," and that "Handles are classes that abstract the users away from the implementation" (col. 232, lines 15-23).

Bowman-Amuah also teaches that a ProxyHandle "is the Handle that clients should use to manage pooled proxies" and that proxy handles must use a particular C++ template "defined by a client template instantiation" to "retrieve the correct proxy" (col. 232, lines 46-52). Thus, Bowman-Amuah teaches a C++ interface for accessing proxies. Bowman-Amuah clearly fails to disclose a first entity accessing the proxy service through messages in a data representation language.

Additionally, **Bowman-Amuah fails to disclose wherein the proxy service appears to the first entity as the second entity**. The Examiner cites, col. 51 of Bowman-Amuah. However, the cited passage does not mention anything about a proxy service appearing to a first entity as a second entity. Instead, the cited passage is part of a larger discussion of implementation considerations including such questions as, “Are changes in data usage anticipated?”, “Is it desirable to shield the user from the data access process?”, “Are available resources an issue?”, “What are the business requirements?” and “Do I already have a component that satisfies this criteria?”.

Bowman-Amuah does not, either at the cited passage or anywhere else describe or otherwise disclose a proxy service appearing as a second entity to a first entity, as recited in Applicants’ claim. Without some specific teaching by Bowman-Amuah regarding a proxy service appearing to a first entity as a second entity, Bowman-Amuah cannot be said to anticipate claim 1.

As such, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 51 and 100.

Regarding claim 136, **Bowman-Amuah fails to disclose a first entity in the first computing environment accessing a proxy service through messages in a data representation language**. The Examiner cites col. 43 of Bowman-Amuah, presumably relying on Bowman-Amuah’s browser extension services and plug-ins. However, as described above regarding claim 1, Bowman-Amuah’s browser extension services and plug-ins do not involve, nor disclose, accessing a proxy service through messages in a data representation language. Furthermore, Bowman-Amuah is silent regarding a first entity accessing a proxy service through messages in a data representation language. In contrast, Bowman-Amuah’s proxy services appear to be accessed via C++ API calls. As noted above regarding claim 1, Bowman-Amuah teaches particular C++ interfaces, classes and templates for creating, storing and accessing proxies. Bowman-Amuah clearly fails to disclose a first entity accessing the proxy service through messages in a data representation language.

Further regarding claim 136, **Bowman-Amuah also fails to disclose wherein the proxy service providing to the first entity an interface to a second entity in the second computing environment comprises providing an advertisement for the second entity, wherein the advertisement for the second entity includes access information for accessing the second entity in the second environment from the first environment**. The Examiner cites col. 56 of Bowman-Amuah. However, the cited passage, as well as the remainder of Bowman-Amuah, fails to mention anything about a proxy service providing to a first entity an interface to a second entity comprising providing an advertisement for the second entity, where the advertisement includes access information for accessing the second entity, as recited in Applicants' claim. Instead, the cited passage describes various aspects of a communications services layer including virtual resources, directory services, messaging and security services.

Bowman-Amuah teaches that virtual resource services "proxy or mimic the capabilities of specialized, network-connected resources" (col. 56, lines 16-21 and col. 57, lines 44-51) but does describe any proxy service providing an advertisement for the second entity, wherein the advertisement for the second entity includes access information for accessing the second entity in the second environment from the first environment.

As such, the rejection of claim 136 is not supported by the cited art and removal thereof is respectfully requested.

In regards to claim 138, contrary to the Examiner's assertion, **Bowman-Amuah fails to disclose a first entity in the first computing environment accessing a proxy service through messages in a data representation language**. As discussed above regarding the rejection of claim 1, Bowman-Amuah is silent regarding a first entity accessing a proxy service through messages in a data representation language. In contrast, Bowman-Amuah's proxy services appear to be accessed via C++ API calls. As noted above regarding claim 1, Bowman-Amuah teaches particular C++ interfaces,

classes and templates for creating, storing and accessing proxies. Bowman-Amuah clearly fails to disclose a first entity accessing the proxy service through messages in a data representation language.

Furthermore, **Bowman-Amuah fails to disclose the proxy service providing to the first entity an interface to a second entity in the second computing environment, wherein said providing an interface comprises sending to the first entity a schema defining one or more messages in the data representation language for accessing the second entity.** The Examiner again cites col. 56. However, the cited passage does not mention anything about a proxy service sending to a first entity a schema defining one or more messages in a data representation language for accessing the second entity. Instead, the cited passage describes various aspects of a communications services layer including virtual resources, directory services, messaging and security services.

Without some specific teaching by Bowman-Amuah regarding sending to the first entity a schema defining one or more messages in the data representation language for accessing the second entity, Bowman-Amuah cannot be said to anticipate Applicants' claim.

For at least the reasons above, the rejection of claim 138 is not supported by the cited art and removal thereof is respectfully requested.

Section 103(a) Rejections:

The Examiner rejected claims 1-5, 19-21, 23, 24, 51-55, 68-70, 72, 73, 100-103, 113, 114, 116 and 117 under 35 U.S.C. § 103(a) as being unpatentable over Tuatini in view of Mead et al. (U.S. Patent 6,061,728) (hereinafter "Mead"), claim 136 as being unpatentable over Tuatini in view of Cheng (U.S. Publication 2001/0032273), Machin et al. (U.S. Publication 2002/0032806) (hereinafter "Machin") and Beck et al. (U.S. Patent 6,604,140) (hereinafter "Beck"), claims 6, 7, 56, 104 and 105 as being unpatentable over Tuatini, Mead and Cheng in view of Beck, claims 12-18, 61-67 and 110-112 as being

unpatentable over Tuatini, Mead, Cheng and Beck in view of Machin, and claims 22, 71 and 115 as being unpatentable over Tuatini in view of AAPA. Applicants respectfully traverse these rejections for at least the following reasons.

Regarding the Examiner's rejection of claims 1-5, 19-21, 23, 24, 51-55, 68-70, 72, 73, 100-103, 113, 114, 116 and 117 under 35 U.S.C. § 103(a) as being unpatentable over Tuatini in view of Mead et al. (U.S. Patent 6,061,728) (hereinafter "Mead"), **the Examiner's rejection is improper because, as shown above, Tuatini is not prior art to the present application.**

Further regarding claim 1, **Tuatini in view of Mead fails to teach or suggest a proxy service providing to the first entity an interface to a second entity in the second computing environment, wherein the proxy service appears to the first entity as the second entity.** The Examiner admits that Tuatini fails to teach a proxy service that provides an interface to a second entity and that appears to a first entity as the second entity and relies upon Mead, citing column 3, line 1 through column 4, line 24 of Mead. Mead teaches a system in which multiple proxy devices coordinate to communicate messages between local area networks via a wide area network using a transparent bridging system. Specifically, Mead teaches the use of a master proxy device that mediates and selects which of the proxy devices should handle messages sent between a local area network and a wide area network.

Moreover, **the Examiner's assertion that Mead teaches the use of a proxy service that appears to a first entity as a second entity is erroneous.** Mead's proxy devices do not appear as other entities to Mead's clients (nor to any other entity), even if combined with the teachings of Tuatini. Nowhere does Mead mention that his proxy devices appear as other entities to components of Mead's system. Instead, Mead's proxy devices route messages received from an end station between two local area networks via a wide area network. Each proxy device routes messages and translates them between an Ethernet protocol and a TCP/IP protocol (Mead, FIG. 3 and column 6, lines 28-60). Mead does not mention that a proxy device appears to the end station as the entity to

which the end station is sending a message, as would be required if Mead were to teach a proxy service that appears to a first entity as a second entity, as the Examiner contends.

The Examiner is apparently relying upon the fact that Mead's system includes a transparent bridging mechanism. However, transparent bridging is well understood in the art and **does not include** a proxy service that provides an interface to a second entity and that *appears as the second entity to a first entity*. Mead's proxy devices are transparent because an entity on one local area network sending a message to another local area network via a wide area network is not aware that the proxy devices are performing the routing. Instead, as noted above, Mead's proxy devices only route network message frames from one network to another. The end stations in Mead's system, even if combined with Tuatini, are not aware of Mead's proxy devices at all and do not view the proxy devices as some other entity in the computing environment.

In response to Applicants' argument above (as presented previously) the Examiner, in the Response to Arguments of the Final Office Action and the Advisory Action, asserts, "Mead's teachings and disclosure are not limited to the applicant[s'] assertions" (underlining by Examiner). The Examiner also repeats the assertion that Mead discloses a proxy service that appears to the first entity as the second entity, citing the same passage (column 3, line 1 – column 4, line 24) as cited in the rejection of claim 1. However, the Examiner fails to make any substantive rebuttal or additional argument regarding the fact that Mead's proxy devices function as a transparent bridging mechanism, and not as recited in claim 1. Nor does the Examiner substantively rebut Applicants' argument that transparent bridging systems do not include proxy services that appear as a second entity to a first entity. Instead, the Examiner refers to a computer dictionary definition of various terms, such as proxy, bridge, schema, etc. However, none of the definitions in the cited reference describe or mention anything about a proxy service that appears as a second entity to a first entity and thus fail to support the Examiner's rejection. Nor does the cited reference describe a transparent bridging mechanism.

Additionally, the Examiner's statements regarding Applicants' claims containing "broadly claimed subject matter" reading on the Examiner's interpretation is clearly incorrect. As noted above (and which the Examiner has failed to properly rebut) Applicants' clearly state that Tuatini in view of Mead fails to teach or suggest a proxy service providing to the first entity an interface to a second entity in the second computing environment, wherein the proxy service appears to the first entity as the second entity. Applicant has demonstrated that Mead's proxy devices, which the Examiner equates to the proxy service of Applicants' claim, do not teach or suggest a proxy service that appears to a first entity as a second entity, even if combined with Tuatini. The Examiner has never provided any interpretation that shows how Mead's proxy devices, which as admitted by the Examiner provide a *transparent* bridging service, can appear as a second entity to a first entity.

Additionally, the Examiner's proposed combination of Tuatini and Mead would not result in a system that includes a proxy service providing to a first entity an interface to a second entity where the proxy service appears to the first entity as the second entity. Instead, the Examiner's proposed combination of Tuatini and Mead would result only in allowing Tuatini's application framework, including the messaging component to also transparently route messages between local area networks via a wide area networks using the multiple proxy devices of Mead. Since neither Tuatini nor Mead, whether considered single or in combination, teaches or suggests a proxy service that appears as another entity, no combination of Tuatini and Mead would include such a proxy service (that appears as another entity).

Moreover, Mead's proxy devices are at a completely different computing layer than Tuatini's messaging component, which the Examiner interprets as the proxy service of Applicants' claim. Tuatini's messaging component does not have anything to do with routing frames between a LAN and a WAN. Even if one were to modify the messaging component of Tuatini, which the Examiner interprets as a proxy service providing to first entity an interface to a second entity, the result would merely allow Tuatini's messaging component to route messages between a local area network and a wide area network and

between an Ethernet protocol and a TCP/IP protocol. Nothing in such a combination would include or suggest that the messaging component would appear as another entity.

Therefore, for at least the reasons presented above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 51 and 100.

Regarding claim 20, **Tuatini in view of Mead fails to teach or suggest where the second environment is a non-message based environment**. The Examiner cites FIG. 41 and paragraphs 122 – 160 of Tuatini. However, the cited passages do not teach or suggest a second environment that is a non-message based environment. The Examiner refers to Tuatini’s system including, “CORBA server or Web Server using other than message based languages”. However the Examiner’s interpretation of Tuatini as well as CORBA and Web Servers is clearly incorrect.

Firstly, Tuatini’s invention is clearly directed towards message-based systems. For example, Tuatini teaches, “[w]hen an application component needs to access functionality provided by remote shared services, the component uses a local *messaging service*” (italics added, Tuatini, Abstract). Additionally, the Examiner’s cited passage describes various aspects of Tuatini’s message passing system. Nothing is mentioned about a non-message based environment.

Secondly, the Examiner’s reference to a CORBA server or Web Server is misplaced. As is well understood in the art, both CORBA servers and Web Servers rely on message-based environments. For example, Tuatini states, “the transport connector will be executed to control the sending of the message to the shared service function and to control the receiving of any response messages”. Tuatini continues by stating, “the transport connector will contain specialized knowledge specific to the transport service (e.g., … **CORBA**, … **HTTP**, … etc.) to be used to communicate with the shared service, such as how to establish a connection and how to send and receive messages” (emphasis added, Tuatini, paragraph [0134]). Additionally, Tuatini teaches, “the transport

connector (or service adapter) then sends the sub-message to the shared service through a connector interface 4115 for that type of shared service 4180 (e.g., ... a CORBA component provided by a CORBA server, ... a Web application provided by a Web server, ...), with the sub-message sent in a manner appropriate to execute the function for that transport service” (emphasis added, Tuatini, paragraph [0137]). Thus, Tuatini clearly teaches that both CORBA and HTTP (e.g. used by Web servers) rely on message-based systems.

Mead, not relied upon by the Examiner in the rejection of claim 20, also fails to teach or suggest a second environment that a non-message based environment. Thus, Mead fails to overcome the above mentioned deficiencies of Tuatini regarding a second environment that a non-message based environment. Therefore, the Examiner’s combination of Tuatini and Mead fails to teach or suggest the limitation of claim 20.

For at least the reasons above, the rejection of claim 20 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 69 and 113.

Regarding the rejection of claim 136, Tuatini in view of Cheng, Machin and Beck fails to teach or suggest that a proxy service providing to the first entity an interface to a second entity in the second computing environment comprises providing an advertisement for the second entity, wherein the advertisement for the second entity includes access information for accessing the second entity in the second environment from the first environment and wherein the advertisement includes information describing one or more computer programming language method calls to methods in the computer programming language provided by the second entity.

The Examiner admits that Tuatini fails to teach this limitation of claim 136. The Examiner cites FIG. 3 and paragraphs 9-12 and 23-24 of Cheng. However, the cited

portions of Cheng do not describe providing an advertisement *including access information and information describing computer programming method calls*. Cheng teaches the use of thin glue layers to bridge a non-IP network with the Internet. Cheng's thin glue layers translate between the IP protocol and the non-IP protocol and allow commands and responses to tunnel between applications in the Internet and the non-IP network (Cheng, paragraph [0011]).

The Examiner seems to be arguing that Cheng's teachings regarding a HAVi (a particular non-IP network) application using a HAVi API to access Internet services implies *providing an advertisement* including access information and describing method calls. However, Cheng does not mention providing any sort of *advertisement that includes access information or describing computer programming language method calls*. Instead, Cheng only refers to the fact that the glue layers can translate between the two protocols. As noted above, the Examiner admits that Tuatini fails to provide an advertisement including access information and describing computer programming language method calls. Thus, Cheng, even if combined with Tuatini, fails to teach providing an advertisement including access information and describing computer programming language method calls. Furthermore, Machin and Beck fail to overcome the above noted deficiencies of both Tuatini and Cheng. Therefore, the combination of Tuatini, Cheng, Machin and Beck fails to teach or suggest a proxy service providing an advertisement including access information and describing method calls.

Moreover, claim 136 corresponds to the subject matter of claim 8 written in independent form. Claim 8 is objected to as being dependent from a rejected base claims, but the Examiner states it would be allowable if rewritten in independent form. Thus, claim 136 should also be allowable.

Thus, for at least the reasons above, the rejection of claim 136 is not supported by the cited art and removal thereof is respectfully requested.

Regarding the rejection of claim 12 as being unpatentable over Tuatini, Mead, Cheng and Beck in view of Machin, the Examiner has stated that claim 8, from which claim 12 depends would be allowable if rewritten in independent form. Thus, since claim 12 depends from claim 8, claim 12 should also be allowable if rewritten in independent form. Similarly, regarding claim 61, the Examiner has stated that claim 57, from which claim 61 depends would be allowable if rewritten in independent form. Thus, since claim 61 depends from claim 57, claim 61 should also be allowable if rewritten in independent form. Thus, claims 12 and 61 should also be allowable if written in independent form.

In regards to claim 16, the Examiner has failed to provide a proper rejection of claim 16. The Examiner rejects claim 16 as part of a rejection of claim 9-18, 59-67 and 107-112. However, the Examiner does not mention anything regarding the limitations of claim 16. Nor does the Examiner cite any portions of Tuatini, Mead, Cheng, Beck or Machine that teaches or suggest the limitations of claim 16. In short, the Examiner has completely ignored the limitations of claim 16, which is clearly improper.

Furthermore, the Examiner's combination of Tuatini, Mead, Cheng, Beck and Machin **fails to teach or suggest providing an advertisement for the stored data to the first entity, wherein the advertisement for the stored data includes access information for the stored data**. Instead, Tuatini teaches that result information is transmitted directly to the requesting client via messages. For instance, Tuatini teaches that a container receives the request messages and forwards them to container adapter and “receives response messages from the container adapter and forwards them to the client system” (Tuatini, paragraph [0065]). Similarly, Tuatini also states, “[a]fter the message has been sent to execute the function of the shared service, the transport connector (or service adapter) then waits to receive response messages (if any) from the function” and that “[f]or each response message received, the transport connector performs the same processing discussed (e.g., translation, or additional security measures) in order to send the response message back to the requesting client” (Parenthesis in original, Tuatini, paragraph, [0138]).

Thus, Tuatini clearly and repeated states that result information is send in message to the requesting client. Nowhere does Tuatini mention providing an advertisement for the stored data to the first entity, where the advertisement includes access information for the stored data. Moreover, as noted above, the Examiner has failed to address the limitations of claim 16 in his rejection.

Thus, for at least the reasons above, the rejection of claim 16 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claim 65.

In regards to claim 17, the Examiner has failed to provide a proper rejection. As with the rejection of claim 16, the Examiner rejects claim 17 as part of a rejection of claim 9-18, 59-67 and 107-112. However, the Examiner does not mention anything regarding the limitations of claim 17. Nor does the Examiner cite any portions of Tuatini, Mead, Cheng, Beck or Machine that teaches or suggest the limitations of claim 17. In short, the Examiner has completely ignored the limitations of claim 17, which is clearly improper.

Furthermore, the Examiner's combination of Tuatini, Mead, Cheng, Beck and Machin **fails to teach or suggest the first entity accessing the advertisement for the stored data and the first entity accessing the stored data in accordance with the access information for the stored data in the advertisement for the stored data.** Instead, as noted above regarding claim 16, Tuatini teaches that result information is transmitted directly to the requesting client via messages. For instance, Tuatini teaches that a container receives the request messages and forwards them to container adapter and “receives response messages from the container adapter and forwards them to the client system” (Tuatini, paragraph [0065]). Similarly, Tuatini also states, “[a]fter the message has been sent to execute the function of the shared service, the transport connector (or service adapter) then waits to receive response messages (if any) from the function” and that “[f]or each response message received, the transport connector performs the same processing discussed (e.g., translation, or additional security measures) in order to send

the response message back to the requesting client” (Parenthesis in original, Tuatini, paragraph, [0138]).

Thus, Tuatini clearly and repeated states that result information is send in message to the requesting client. Nowhere does Tuatini mention providing an advertisement for the stored data to the first entity, where the advertisement includes access information for the stored data. Moreover, as noted above, the Examiner has failed to address the limitations of claim 17 in his rejection.

Thus, for at least the reasons above, the rejection of claim 17 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claim 66.

Regarding the rejection of claim 22 as being unpatentable over Tuatini in view of AAPA, the Examiner states that it would have been obvious to combine the teachings of Tuatini and the AAPA because the AAPA’s use of Jini environment would provide access to the Jini services. Applicants disagree with the Examiner’s statement. The Examiner’s statement is entirely conclusory. Applicants submit that such a broad conclusory statement does not provide a sufficient reason to combine the teachings Tuatini and the AAPA. “The factual inquiry whether to combine references must be thorough and searching.” *McGinley v. Franklin Sports, Inc.*, 60 USPQ2d 1001, 1008 (Fed. Cir. 2001). It must be based on objective evidence of record. “This precedent has been reinforced in myriad decisions, and cannot be dispensed with.” *In re Lee*, 61 USPQ2d 1430, 1433 (Fed. Cir. 2002).

The Federal Circuit has stated: “[o]ur case law makes clear that the best defense against the subtle but powerful attraction of a hindsight-based obviousness analysis is rigorous application of the requirement for a showing of the teaching or [reason] to combine prior art references.” The need for specificity pervades this authority. See, e.g., *In re Kotzab*, 217 F.3d 1365, 1371, 55 USPQ2d 1313, 1317 (Fed. Cir. 2000) (“particular findings must be made as to the reason the skilled artisan, with no knowledge of the

claimed invention, would have selected these components for combination in the manner claimed” (emphasis added)); *In re Rouffet*, 149 F.3d 1350, 1359, 47 USPQ2d 1453, 1459 (Fed. Cir. 1998)

The Examiner has failed to provide any reason for modifying Tuatini in view of AAPA. Instead the Examiner has merely pointed to standard boilerplate text indicating that Tuatini’s system may be modified, but that does not provide any reason for the specific modification suggested by the Examiner. Similarly, nothing from AAPA provides any reason to modify the teaching of Tuatini to include the Jini environment.

Just because the Jini environment was known in the prior art, does not mean that one of ordinary skill in the art would modify the teachings of Tuatini with the Jini environment. The Examiner has provided no objective evidence of record to the contrary. The Examiner has only shown that both Tuatini and the Jini environment were known in the art. However, the Examiner’s stated reason, namely, “to utilize Jini services of the Jini environment so that a client will be able to access [sic] advertisement related information from the remote servers of the Jini network through proxy services” amounts to nothing more than a conclusory statement based in hindsight analysis of the present application.

As noted above, the Examiner has merely referred to features of the Jini environment, such as “[t]he Jini environment would provide access to Jini services”, “Jini services would provide information to the client over the network” and “[t]he client would utilize the provided information”. Thus, the Examiner’s stated reason amounts to nothing more than concluding that since both Tuatini’s system and the Jini environment were known it would be obvious to combine them, which as noted above, is clearly improper.

In light of the above remarks, Applicants assert that the rejection of claim 22 is not supported by any evidence of record. Withdrawal of the rejection is respectfully

requested. Similar remarks as discussed above in regard to claim 22 apply to claims 71 and 115.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-72200/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Applicants

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: January 21, 2008